LA-UR-16-26530
Approved for public release; distribution is unlimited.

Title: Introduction to Numerical Methods

Author(s): Schoonover, Joseph A

Intended for: Lecture for Parallel Computing Summer Research Internship at the National Security Education Center Online public release of talk slides.

Issued: 2016-08-25

Introduction to Numerical Methods

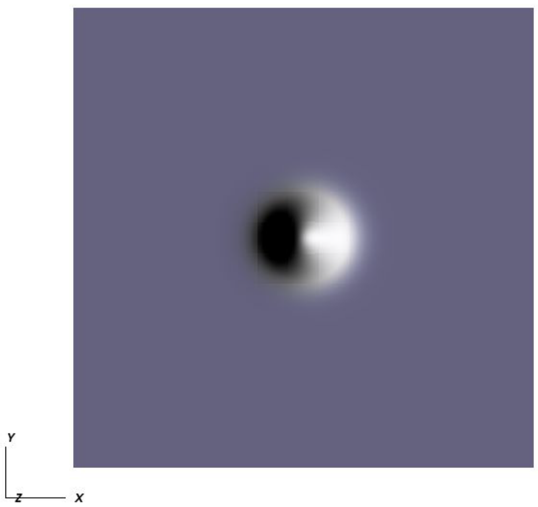
Joseph Schoonover

June 14th, 2016

Numerical Methods

Los Alamos NATIONAL LABORATORY EST. 1943

Repetitive algorithms to obtain approximate solutions to mathematical problems

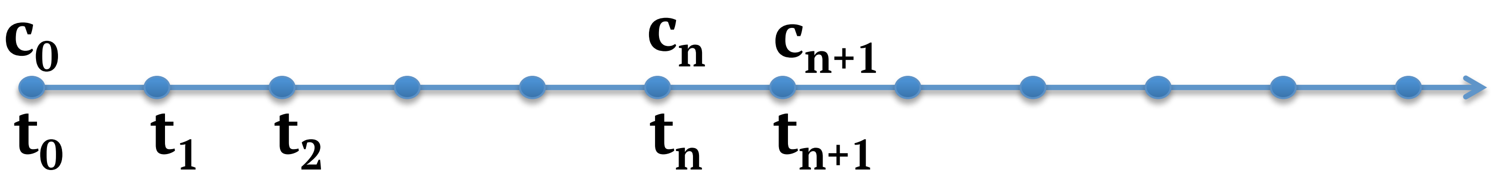


Sorting
Searching
Root Finding
Optimization
Interpolation
Extrapolation
Least squares regression
Eigenvalue Problems
Ordinary Differential Equations
Partial Differential Equations

Starting Simple

$$c_t = \lambda c$$

Goal : Given an initial condition, use basic arithmetic to estimate the solution at a later time.

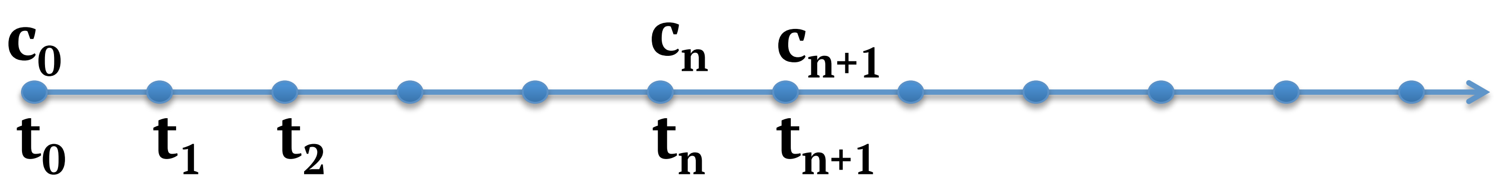


Continuous to Discrete

$$c_t = \lambda c$$

Use Taylor Series

$$c_t|_{t^n} = \frac{c^{n+1} - c^n}{\Delta t} + \frac{\Delta t}{2} c_{tt}|_{\xi}, \quad \xi \in [t_n, t^{n+1}]$$



Introducing Errors

Los Alamos NATIONAL LABORATORY EST. 1943

$$c^{n+1} = c^n + \lambda \Delta t c^n - \underbrace{\frac{\Delta t^2}{2} c_{tt}|_\xi}_{TruncationError}$$

Approximate the solution by dropping the truncation error

"Forward Euler"

$$\tilde{c}^{n+1} = \tilde{c}^n + \lambda \Delta t \tilde{c}^n$$

How big can the time step be ?

Time Step and Stability

How can we control the error growth ?

Error

$$E^n = c^n - \tilde{c}^n$$

$$E^{n+1} = \underbrace{(1 + \lambda \Delta t)}_{Amplification} \; E^n - \frac{\Delta t^2}{2} c_{tt}|_\xi$$

The truncation error acts as a source

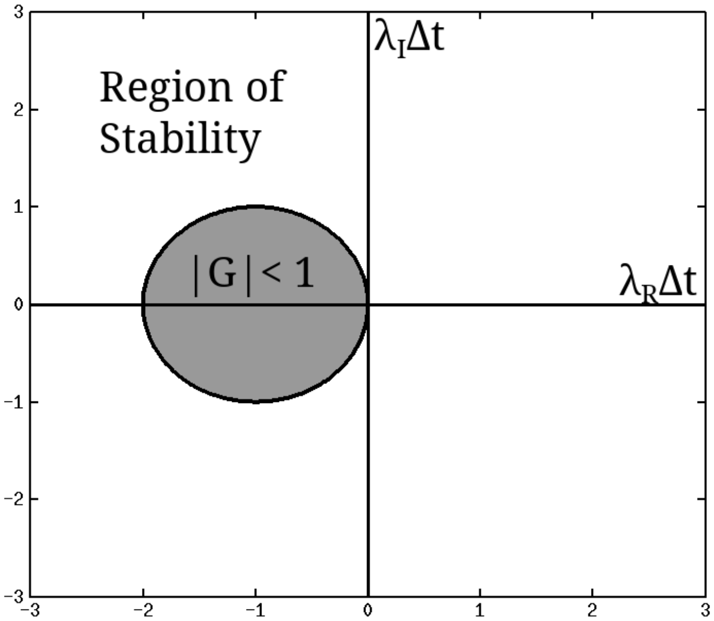
Error can be amplified!

Time Step and Stability

How can we control the error growth ?

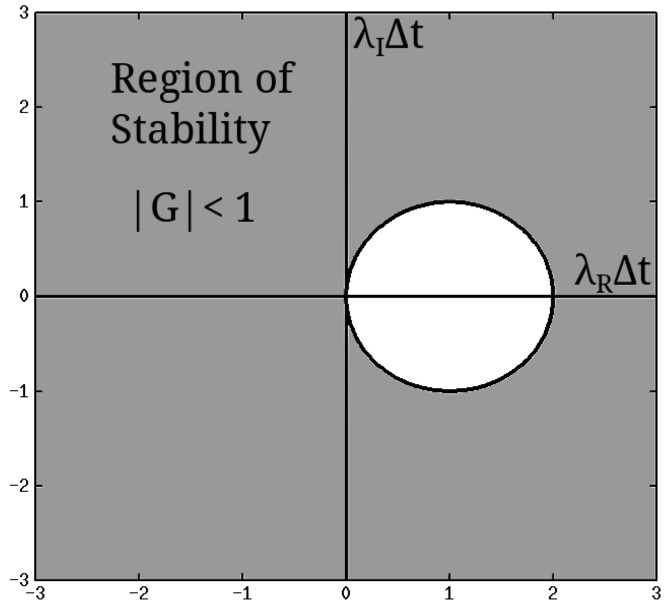
<u>Don't amplify the error!</u>

$$|G| = \left|\frac{\tilde{c}^{n+1}}{\tilde{c}^{n}}\right| = \underbrace{|1 + \lambda\Delta t|}_{ForwardEuler} < 1$$



Stability Diagrams and Implicit Integration Schemes

$$|G| = \underbrace{\left|\frac{1}{1-\lambda\Delta t}\right|}_{BackwardEuler} < 1$$



*Implicit schemes allow for larger time steps

Convergence

If the truncation error is $\mathcal{O}(\Delta t^n)$
and the integration scheme is stable,
then the approximate solution will converge
to the exact solution as $\Delta t \to 0$
at a rate proportional to Δt^n.

Systems of ODEs

PDE

$$\vec{s}_t + \nabla \cdot \vec{f} = \vec{q}$$

Spatial Discretization →

ODE System

$$\vec{c}_t = \boldsymbol{A}\vec{c}$$

Stability

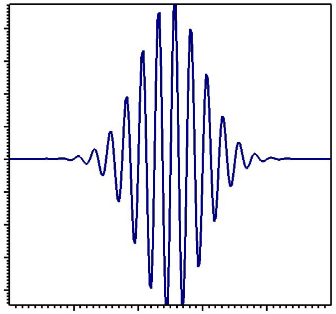
- Decoupling = Diagonalization

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$$

- Maximum eigenvalue determines time step restrictions

$$(w_i)_t = \lambda_i w_i$$

Finite Difference Method
Upwind Advection



$$c_t + uc_x = 0$$

Use Taylor Series for the spatial derivative

Restrict the truncation to zero at each node

$$(\tilde{c}_i)_t = -\frac{u}{\Delta x}(\tilde{c}_i - \tilde{c}_{i-1})$$

Finite Difference Method

Upwind Advection

$$\vec{c}_t = \boldsymbol{A}\vec{c}$$

Spatial discretization results in a sparse ODE system;

$$A_{i,i} = \frac{-u}{\Delta x},\; A_{i,i-1} = \frac{u}{\Delta x}$$

Integration by Forward Euler

$$\tilde{c}_i^{n+1} = \tilde{c}_i^n - \frac{u\Delta t}{\Delta x}(\tilde{c}_i^n - \tilde{c}_{i-1}^n)$$

Finite Difference Method
Upwind Advection, Group Example

$$\tilde{c}_i^{n+1} = \tilde{c}_i^n - 0.5(\tilde{c}_i^n - \tilde{c}_{i-1}^n)$$

You each are assigned a process ID and three nodes

<u>To advance the system</u>

1. Exchange data with your "upstream neighbor"
2. Update the solution at your three nodes
3. Do your file I/O.
4. Wait for your neighbor to finish.
5. Repeat

Sample Solution Output

Solution.001.00000.data

i	c
1	0.000
2	0.000
3	0.000

Finite Difference Method

Implicit Upwind Advection

$$\tilde{c}_i^{n+1} = \tilde{c}_i^n - \frac{u\Delta t}{\Delta x}\underbrace{(\tilde{c}_i^{n+1} - \tilde{c}_{i-1}^{n+1})}_{\text{Evaluated at the next time step}}$$

Evaluated at the next time step

Forward Integration requires matrix inversion

$$(\boldsymbol{I} - \Delta t \boldsymbol{A})\vec{c}^{n+1} = \vec{c}^{n}$$

General Discretization Overview

- Approximate continuous functions by discrete observations
- Approximate derivative operators by difference operators

PDE

$$\vec{s}_t + \nabla \cdot \vec{f} = \vec{q}$$

Spatial Discretization →

ODE System

$$\vec{S}_t + \boldsymbol{A}\vec{S} = \vec{Q}$$

- Apply a suitable time integrator

Discretization Flavors

Mesh-based

Strong (Collocation)	Weak (Galerkin)	Conservative
Finite Difference	Finite Element	Finite Volume
Pseudo-Spectral	Spectral	
	Spectral Element	

Mesh-free

Smoothed Particle Hydrodynamics
Element-free Galerkin
Discrete Vortex Method
Radial Basis Methods

Stability and Work
Explicit Schemes

- Wall-Time ~ (Number of Floating Point Operations) /(FLOP/sec)
- Remember, stability requirements impose time-step restrictions

$$\lambda_{max}\Delta t < C$$

- We want to integrate to a known time, which requires M time-steps

$$M = \frac{T}{\Delta t}$$

- The number of floating point operations per time step is proportional to the number of degrees of freedom

$$W_{float} \propto \frac{NT\lambda(N)}{k_{FLOPS}}$$

Summary

- I showed too many equations.
- I am sorry.
- Discretizations
  - allow us to approximate solutions to mathematical models of physical systems using a repetitive algorithm.
  - introduce errors that can lead to numerical instabilities if we are not careful.
- We barely scratched the surface of numerical methods.
- Let's talk more!